(2)

AD

TECHNICAL REPORT ARCCB-TR-92031

PACRAT: A MACHINE BRAIN

PART ONE

RAYMOND SCANLON
MARK JOHNSON

JUNE 1992

US ARMY ARMAMENT RESEARCH,
DEVELOPMENT AND ENGINEERING CENTER
CLOSE COMBAT ARMAMENTS CENTER
BENÉT LABORATORIES
WATERVLIET, N.Y. 12189-4050

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The use of trade name(s) and/or manufacturer(s) does not constitute an official indorsement or approval.

DESTRUCTION NOTICE

For classified documents, follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

For unclassified, unlimited documents, destroy when the report is no longer needed. Do not return it to the originator.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1992

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
PACRAT: A MACHINE BRAIN
PART ONE

5. FUNDING NUMBERS
AMCMS: 6111.01.91A0.0
PRON: 1A1AZ11BNMBJ

6. AUTHOR(S)
Raymond Scanlon and Mark Johnson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army ARDEC
Benet Laboratories, SMCAR-CCB-TL
Watervliet, NY 12189-4050

8. PERFORMING ORGANIZATION REPORT NUMBER
ARCCB-TR-92031

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army ARDEC
Close Combat Armaments Center
Picatinny Arsenal, NJ 07806-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
A computer program to simulate and exercise a mammalian brain has been written. We describe the section that builds the brain.

14. SUBJECT TERMS
Artificial Intelligence, Nonliving Intelligence, Machine Intelligence, Thinking Machine, Intelligence

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

INTRODUCTION

The vertebrate brain, and in particular the mammalian brain, consists of:

1. a neural net, the cortex, that responds differentially to the exterior universe,

2. motor programs hardwired in the basal ganglia,

3. a throttle, the thalamic reticular nucleus, that can stop the flow of signal energy from the sensory neurons to the neocortex, and also the flow of motor programs from the basal ganglia to the motor cortex,

4. an averager, the cerebellum, that keeps running averages of the signals sent to the muscles and incoming signal information.

Associated with this brain is a mind. The association is not part of the universe of experience.

BACKGROUND

All of the neurons in the central nervous system are equal; but some are more equal than others. What are the essentials? We need a nerve net that responds differentially to the universe. Response to oriented line segments and the orthogonal movement of these segments have been identified (Hubel, 1988). We need a set of motor programs provided by the genes and we nominate the basal ganglia to provide them. We need an interruption mechanism to disconnect the motor program from the motor neuron. The thalamic reticular nucleus is a likely candidate. Lastly we need modifiable synapses. The Hebbian hypothesis currently serves. If a synapse participates in firing a neuron, it is strengthened.

By motor program we mean a hardwired neural net in which neuron A excites neuron B and 10 msec later neuron C. Neuron B and C are connected to motor neurons so that one muscle fiber contracts 10 msec before the other and a delicate movement is produced. The essential quality of the motor program is that it is hardwired. The brain does not "recall" the motor program; the signal energy triggers it. Also, we speak of two neurons rather than 2000.

We cannot see the mind as object. Nothing rational can be said of the mind. It is the subject for which the world is object.

The guiding question for this endeavor is: Will this notion advance the design of an electronic brain?

The Unknowable

We look at consciousness from the outside, we see an animal that is alert. When I look inside, I am aware. I need not think; to be aware is enough. The brain and the mind may be the same thing, may be different aspects of the same thing, may be totally different things. We do not know; we cannot know. Man cannot know the relationship of 'alert' to 'aware.'

Mind-Body

The mind is aware; the brain is alert.

In the beginning, the mind watched; the brain thought and the mind was aware of the thought. Awareness is of the mind. This quality of awareness is a mystery for which we have no explanation.

I am aware.
You are alert.
He exhibits intelligent behavior.

This is the box in which we live. Is there a way out? We doubt it. Although we gladly admit all vertebrates as potentially aware--certainly all mammals (Kissin, 1986:81) and probably octopi and squid--we have never found an experiment that gives any hint of testing for awareness. Turing's test tells us nothing; he speaks only of intelligence.

All brain activities reduce to a pattern of connection. All mental events reduce to awareness of neural activation.

Any animal that is alert may be aware.
A machine may be alert.
Any animal that has plastic synapses may learn.
A machine may have plastic synapses.

The metalanguage differs from rational language. We may not use rational language to talk about minds. Minds only experience rationalism. We may use rational language to talk about brains.

We are under no direction to speak only rationally.

Awareness

Consciousness of the outer world occurs at points. One neuron (or one glomerulus) in the lateral geniculate nucleus, rising to awareness, is enough. The mind is aware of the activity of single neurons. We readily sense a point source of light that excites, at most, only a few neurons. The mind is not equally aware of all neurons. Some neurons rank high, whereas other neurons are silent.

The Homunculus

The homunculus is a little man who sits in the middle of the brain watching a television screen and pushing buttons. The homunculus stores memories, retrieves them, and compares them with present experience. We exorcise the homunculus by postulating a passive brain. The mind watches the signal energy coursing through the brain, but the brain knows nothing. It neither stores, nor retrieves, nor compares. It is wholly passive. We need a valve to shut off the flow of motor output temporarily. The thalamic reticular nucleus is in the right place and is capable of doing it. "What function the reticular nucleus really effects is also largely a matter of conjecture" (Jones, 1985:819).

The Passive Brain

We see a brain composed of neurons and the neurons as generators of electrical pulses. Signal energy courses through this passive brain. There is no need for rigor here. We say most of them generate pulses, with a minority that instead deals in graded potentials. We know enough about the brain to imagine how it works, but nowhere near enough to do anything but guess.

We may view the brain as electronic circuitry. We can move the abstract circuitry from one substrate to another, from organic structure to silicon.

Thought

E. V. Evarts quoted Walle Nauta as having said "What is a thought except a movement that is not connected to a motor neuron" (Ciba Foundation, 1984:272).

Schneider thought Nauta might be stretching the point slightly; we think not (Schneider, 1987:7). We think it is a simple, straightforward statement that needs no qualification. The manner of the disconnection is what we must examine. Just how do we disconnect the motor program from the motor neuron in a passive brain? The disconnection is not physical, of course...no axon dangles in space. It must be done electrically, by inhibition. The only nucleus that is in a position to do this is the reticular nucleus of the thalamus. It is in position to interrupt the flow from the basal ganglia to the motor cortex.

Thought has two faces. One is the mechanical action of the brain; the other is the awareness of this action. This awareness is that which philosophers have traditionally called thought. It is in this talk that we find mental images and logical forms. We have no need of these notions. The style of thinking that interests us is that of Nauta. We would like to say it is the brain that thinks, with the mind as spectator only. The mind is aware of thinking as it is of pain and happiness.

Thinking follows when the reticular nucleus of the thalamus inhibits sensory input. The signal energy that has already passed through is then free to vibrate and oscillate. We present thinking as a mindless activity of the brain of which the mind is only aware. This we see as the best way to approach the design of an electronic brain. Extended thought is a luxury of a dominant species at the top of the food chain, an aberration, deviant. Thought has marginal utility.

THE PROGRAM: PART ONE

The program, Pacrat, contains a brain, an organism, and an environment in which the organism moves about as directed by the brain. The program splits logically into three parts. The first defines the brain, the organism, and the universe. The second exercises the brain. The third moves Pacrat and evaluates the response of the universe. A subsidiary program, running on a workstation, provides the graphic output that allows us to judge the actions of Pacrat. Here we describe the first part of the main program.

COMMON

We designed the program about named common. The common blocks completely define the condition of the brain, the stance of the organism, and the situation of the environment. The program manipulates the elements in the named common. An appendix sets forth their structure.

MAIN

The main program calls a subroutine, SETUP, that builds a brain and an environment. It then calls CYCLE in a do loop. CYCLE steps through the thinking, movement, and the reaction of the environment.

```
program pacrat
include 'brain'
chkout=.false.
call setup
call lister
do 20 j=1,kbgend
do 10 k=1,kltend
call cycle
```

```
      call chflag
      if(mod(icnt,ksnap).eq.0) call hotout
10    continue
20    continue
      call closer
      call exit
      end

      subroutine cycle
      include 'brain'
      call updacl
      call ras
      call update
      if(mod(icnt,10).eq.0) then
c      call weight
      end if
      if(.false.) then
c      if(mod(icnt,100).eq.0) then
      write(*,*) 'icnt',icnt
      call growth
      call recept
      call nusyns
      end if
      if(mod(icnt,10).eq.0) then
      call mover
      call univrs
      if((icnt.ge.0).and.(icnt.le.100))then
      call actvty
      end if
      end if
      return
      end
```

SETUP

The first call of the main program is to SETUP. SETUP opens all the working files. The working routines open the checkout files where the action occurs.

brain.nms	contains the names of the brain centers
brain.cnx	specifies the efferent connections and the scheme of connection
walls	the parameters of the universe
trail	history of Pacrat's progress
brain.ckr	
brain.strt	parameters for a given run

brain.snap	the entire contents of the brain for a hot start

Following the opening of files, we read in a record from brain.strt containing starting parameters. The following variables define the run:

chkout	a checkout flag
hotstr	true calls for a hot start; we read in a brain snapshot
ksnap	an interval (in cycles) between between brain snapshots
kaix	a starter for the random number generator
kbgend	terminator for the outer loop in the main program
kltend	terminator for the inner loop

Next, we call RANDIT(kaix) to initialize the random number generator. We use the random numbers to avoid the artificial regularity inherent in the serial makeup of Fortran do loops. If two or more synapses are in the same condition, we add a small random increment so that neurons will be selected in an arbitrary order.

Next we read in information about desired check out using the namelist method. The namelist contains:

flags (100)	(logical) these flags control the check out listings
lstnr(10)	the number of entries in nuclst
nuclst(100)	(character*30) the names of the nuclei that we want to examine
kbndls(3)	these numbers control the do loop that lists the status of neurons

At this point we decide, depending on the truth value of hotstr, whether to make a hot start by calling HOT. The subroutine HOT will read in the contents of the brain as it was at the last snapshot. Alternatively we call COLD to build a brain from parametric information.

```
      subroutine setup
      include 'brain'
      namelist/snaps/ flags,lstnr,nuclst,kbndls
c
c all working files are opened with the exception of
c brain.snap that will be opened in subroutine hot if reading
c or writing a snapshot of the brain is called for.
c
      open(1,file='brain.nms',status='old')
      open(2,file='brain.cnx',status='old')
```

```
      open(3,file='walls',status='old')
      open(4,file='trail')
      open(8,file='brain.ckr')
      open(9,file='brain.strt',status='old')
c
c 11,12,13, and 18 are check-out files.
c
      open(20,file='radii',status='old')
      read(9,10) chkout,hotstr,ksnap,kaix,kbgend,kltend
      write(*,*) chkout,hotstr,ksnap,kaix,kbgend,kltend
   10 format(2l5,4i10)
c
c     chkout  a checkout flag
c     hotstrt if true a hot start is called for, a brain
c             snapshot will be read in.
c     ksnap   the interval at which brain snapshots will be taken.
c     kaix    a starter for the random number generator.
c     kbgend  outer loop in main
c     kltend  inner loop in main
c
c the random number generator is initialized.
c
      aix=randit(kaix)
      read(9,snaps,end=35)
   35 continue
c
c a selection of flags and bounds are read in. These are used
c to print out states of the brain during program execution.
c
c     flags(100)  logical     these are flags used to direct
c                             monitoring.
c       flags(1)  In subroutine chflag, if true sends control to
c                 chekpr.
c
c     lstnr(10)
c       lstnr(1)  number of entries in nuclst.
c
c     nuclst(100) character*30
c       a list on nuclei to be printed out as they are called for.
c
c     kbndls(3)
c       kbndls(1), kbndls(2), and kbndls(3) form the parameters
c       of a do loop in chekr.
c
c
c select either a hot or a cold start.
c
      if(hotstr) then
      call hot
      else
      call cold
      end if
```

```
c
c initialize hubel
c
      call hublin
c
c initialize visual
c
      call visiin
c
c initialize motr
c
      call motrin
      return
      end

      function randu(x)
      save
      iy=iy*65539
      if(iy.lt.0) iy=iy+2147483647+1
      randu=iy*.4656613e-9
      return
      entry randin  (ix)
      iy=ix/2
      iy=iy*2+1
      randin  =0.0
      return
      end

      function gauss(sigma,amean)
      dimension f(10,10)
      save
      a=0.0
      do 50 i=1,12
   50 a=a+randu(z)
      gauss=(a-6.0)*sigma+amean
      return
      entry randit(ix)
      call randin(ix)
      do 60 j=1,10
      do 60 k=1,10
   60 f(j,k)=randu(z)
      randit=0.0
      return
      entry rand(x)
      j=10.0*randu(x)+1.0
      k=10.0*randu(x)+1.0
      swap=f(j,k)
      f(j,k)=randu(x)
      rand=swap
      return
      end
```

HOT

This routine reads in the contents of the named commons from a snapshot file that we established in a previous run by a call to HOTOUT. This restarts Pacrat as if there had been no intermission.

HOTOUT

HOTOUT opens a file called brain.snap and dumps the entire contents of the named commons. This large file contains all the numbers and logical variables that define the brain, the organism, and the environment. These are the synaptic connections with their facilitation and conditions: a history of the neurons, and all the numbers that make for a simulation; the description and location of the organism; and a description of the universe.

```
      subroutine hot
c
c when we rewrite this for new common, watch out for
c line length.
c
      include 'brain'
      open(10,file='brain.snap',status='old',
     1    form='unformatted')
      read(10) a,b,centrs,nuclei,nrctrs,nrnucs,centrs,
     1    nuclei,kcnucs,kruse,krnet,krgro,ksens,nucnet,
     2    nucgro,kcensz,knucsz,knucwd,stinc,knucs,kside
      read(10) ksynps
      read(10) potent
      read(10) synred
      read(10) ksynct,histry
      read(10) ktime
      read(10) alertd,runavg,thresh,recptv,grfctr,synare,grinc,
     4    icnt,reduce,frustr,frlvla,frlvlb,rewavg,recovr,
     5    thrup,thrdwn
      read(10) chkout,hotstr,ksnap,knrrad,radii,body,nrwall,
     6    awalls,avectr,amoton,bmoton,amovr,phase,radi,
     7    snpass,ampass,dist,dxm
      close (10)
      return
      entry hotout
      open(10,file='brain.snap',form='unformatted')
      write(10) a,b,centrs,nuclei,nrctrs,nrnucs,centrs,
     1    nuclei,kcnucs,kruse,krnet,krgro,ksens,nucnet,
     2    nucgro,kcensz,knucsz,knucwd,stinc,knucs,kside
      write(10) ksynps
      write(10) potent
      write(10) synred
      write(10) ksynct,histry
      write(10) ktime
      write(10) alertd,runavg,thresh,recptv,grfctr,synare,grinc,
     4    icnt,reduce,frustr,frlvla,frlvlb,rewavg,recovr,
     5    thrup,thrdwn
      write(10) chkout,hotstr,ksnap,knrrad,radii,body,nrwall,
     6    awalls,avectr,amoton,bmoton,amovr,phase,radi,
```

```
7       snpass,ampass,dist,dxm
      close (10)
      return
      end
```

COLD

This routine schedules all the routines that build the brain. It calls in turn INIT, INITX, NAMER, NET, and CONNEX. After these calls, the routine returns control to SETUP.

```
      subroutine cold
c
c This routine creates a brain by calling the following
c subroutines.
c
      include 'brain'
      call init
      call initx
      call namer
      call net
      call connex
      return
      end
```

INIT

INIT establishes several parameters, and resets the cycle count, icnt. Then it defines certain bounding values:

memsiz	an upper bound on the number of neurons
kcensz	an upper bound on the number of centers
knucsz	an upper bound on the number of nuclei
kside	the number of neurons on a side of a nucleus
knucwd	an upper bound on the number of nuclei to which this nucleus may have efferent connections, either genetic or epigenetic

The routine establishes parameters about the dynamic action of the brain:

thrup	increment of threshold hyperpolarization
thrdwn	increment of threshold hypopolarization
reduce	decrement of synaptic reserve

recovr	increment of synaptic reserve
stinc	synaptic strength at genetic establishment
grinc	synaptic strength at epigenetic growth
peaked	(a skeleton)
frlvla	upper level of frustration
frlvlb	lower level of frustration
dxm	a step size for movement of the organism
phase	original value for sinusoidal movement

The routine returns control to COLD.

```
      subroutine init
      include 'brain'
c
c Certain basic parameters are declared.
c         icnt        the cycle counter
c         kcensz      an upper limit on the number of brain centers
c         knucsz      an upper limit on the number of nuclei
c         kside       the number of neurons on one side of a nucleus.
c                the nuclei are square.
c         knucwd          an upper limit on the number of nuclei that the
c             neurons in this nucleus can be efferent upon or
c             grow to.
c         thrup       hyperpolarization increment
c         thrdwn      hypopolarization decrement
c         reduce      decrement in synaptic availability
c         recovr      increment in synaptic availability
c         stinc       genetic synaptic potentiation
c         grinc       epigenetic synaptic potentiation
c         frlvla      upper check on frustration (frustr)
c         frlvlb      lower check on frustration
c         dxm    a step size for movement of organism
c         phase       an initial value for sinusoidal movement
c
      icnt=0
      kcensz=100
      knucsz=300
      kside=16
      knucwd=20
      thrup=0.003
      thrdwn=0.001
      reduce=0.003
      recovr=0.001
      stinc=0.2
```

```
      grinc=0.01
      frlvla=0.9
      frlvlb=0.1
      dxm=0.05
      phase=0.0
      return
      end
```

INITX

This routine resets all the arrays that are to contain the dynamic history of the brain. These are: kruse, krnet, krgro, nucnet, nucgro, ksynct, thresh, and histry. We initialize runavg at 0.5 and reset alertd, ksens, and snpaso. Then we read in a description of the universe into awalls from the file walls, and place the starting position of the organism in amovr. We place the position of the head of the organism in avectr and amoton, and reset the working array, bmoton.

Next we read in a description of the organism from the file radii and place it in the array, radii.

We return control to COLD.

```
      subroutine initx
      include 'brain'
      do 5 j=1,knucsz
      kruse(j)=0
      krnet(j)=0
      krgro(j)=0
      do 5 k=1,knucwd
      do 5 l=1,4
      nucnet(j,k,l)=0
    5 nucgro(j,k,l)=0
      do 10 l=1,knucsz
      do 10 k=1,kside
      do 10 j=1,kside
      ksynct(j,k,l)=0
      thresh(j,k,l)=0.0
   10 continue
      do 110 l=1,knucsz
      do 110 k=1,kside
      do 110 j=1,kside
      do 105 i=j,10
      histry(i,j,k,l)=0.0
  105 continue
  110 continue
      do 120 i=1,knucsz
      runavg(i)=0.5
      alertd(i)=0.0
  120 continue
      read(3,25) nrwall
   25 format(i5)
      do 30 k=1,nrwall
      read(3,27) ((awalls(i,j,k),i=1,3),j=1,2)
   27 format(6f10.5)
```

```
   30 continue
      close(3)
      do 45 k=1,100
      do 44 j=1,3
      amovr(j,k)=0.0
   44 continue
c
c set following for initial displacement of pacrat.
c
      amovr(1,k)=7.0
c      amovr(3,k)=5.0
   45 continue
      do 46 j=1,3
      avectr(j)=0.0
      do 43 i=1,3
      amoton(i,j)=0.0
      if(i.eq.j) amoton(i,j)=1.0
   43 continue
   46 continue
c
c special coding to start at a 45 degree angle.
c
      arg=3.141593/4.0
      amoton(1,1)=cos(arg)
      amoton(3,3)=cos(arg)
      amoton(1,3)=-sin(arg)
      amoton(3,1)=sin(arg)
c
      do 90 l=1,100
      do 90 k=1,3
      do 90 j=1,3
      bmoton(j,k,l)=0.0
      if(j.eq.k) bmoton(j,k,l)=1.0
   90 continue
      read(20,47)knrrad
   47 format(i5)
      do 60 k=1,knrrad
      read(20,48) (radii(1,j,k),j=1,3,2)
   48 format(2f10.5)
      radii(1,3,k)=radii(1,3,k)/3.0
      radii(1,2,k)=0.0
      radii(2,3,k)=-radii(1,3,k)
      do 50 j=1,2
      radii(2,j,k)=radii(1,j,k)
   50 continue
   60 continue
      return
      end
```

NAMER

This routine reads in the names of the centers and nuclei that are to form the brain. We verify that no name duplicates a previously defined name and store the center in centrs and the nuclei in nuclei. We place the count of centers in nrctrs and the count of the nuclei in nrnucs, then return control to COLD.

```
      subroutine namer
      include 'brain'
c
c it is assumed that all the names of centers and of nuclei
c will be read in. there will be a center and a nucleus on
c each card. the centers will appear in groups, and no nucleus
c will be repeated.
c
c why the following appears is not known
c
      kadhoc=77
c
      kctrs=1
      knucs=1
      read(1,20,end=100) a,b
      centrs(kctrs)=a
      nuclei(knucs)=b
      kcnucs(kctrs,1)=knucs
c
c this is the main loop.
c
   10 continue
      read(1,20,end=100) a,b
   20 format(2a30)
c
c check that the nucleus is unique
c
      call ident(b,jn,kn)
      if((jn.ne.0).and.(kn.ne.0))then
      write(13,35) a,b
   35 format(1h ,'dupe ',2(a30,2x))
      end if
c
c is this a new center
c
      if(lge(centrs(kctrs),a).and.lle(centrs(kctrs),a))then
      knucs=knucs+1
      if(knucs.gt.knucsz) then
      write(13,45) knucs,knucsz
   45 format(1h ,'exceeded knucsz in namer',2i10)
      call exit
      end if
      nuclei(knucs)=b
      else
c
c it is.
```

```
c
      kcnucs(kctrs,2)=knucs
      kctrs=kctrs+1
      knucs=knucs+1
      if((knucs.gt.knucsz).or.(kctrs.gt.kcensz)) then
      write(13,55) kctrs,kcensz,knucs,knucsz
   55 format(1h ,'exceeded kcensz or knucsz in namer',4i10)
      call exit
      end if
      kcnucs(kctrs,1)=knucs
      centrs(kctrs)=a
      nuclei(knucs)=b
      end if
c
c nrctrs and nrnucs are to contain the total of centers and nuclei.
c
      kcnucs(kctrs,2)=knucs
      go to 10
c
c continue the main loop.
c
  100 continue
      nrctrs=kctrs
      nrnucs=knucs
      do 120 k=1,nrctrs
      write(18,110) k,kcnucs(k,1),kcnucs(k,2)
  110 format(3i5)
  120 continue
      return
      end
```

IDENT

This routine returns the center, j, and the nucleus, k, of the name, a. If the name does not appear in the list of nuclei, we reset j and k.

```
      subroutine ident(a,j,k)
      include 'brain'
c
c this routine depends upon nuclei appearing only once.
c
      do 40 ja=1,nrctrs
      do 30 ka=kcnucs(ja,1),kcnucs(ja,2)
      if(lle(a,nuclei(ka)).and.lge(a,nuclei(ka)))then
      j=ja
      k=ka
      return
      end if
   30 continue
   40 continue
      j=0
```

```
      k=0
      return
      end
```

NET

NET reads in a description of the neural connections that define the brain: lx, ly, lw, lz, a, and b, where a is the name of the nucleus in question and b is the name of the nucleus to which it is afferent. Of course, each nucleus may be afferent to more than one other nucleus, including itself.

We check a and b to see that they are present in the list of nuclei in nuclei. We do this check in IDENT and then record the following distinctions:

lx=1	this is a reservation of space
lx=2	genetic instructions follow
lx=3	epigenetic instructions follow

If lx=1, then ly is the square root of the number of neurons to be reserved for this nucleus. If lx=2 or 3, then ly is the number of efferent synapses for each neuron in this nucleus. If ly=10, then the first digit identifies the scheme of connection (other than topological).

If lw=3, this is a sensory nucleus. If lw=4, this is a motor nucleus.

We use lz to make a distinction between ipsilateral and contralateral synapses, and between excitatory and inhibitory connections.

lz=1	synapses are ipsilateral and excitatory
lz=2	synapses are ipsilateral and inhibitory
lz=3	synapses are contralateral and excitatory
lz=4	synapses are contralateral and inhibitory

The call to IDENT returned a value, kn, that is the identifying number of this nucleus. A second call to IDENT returns the value, kna, that identifies the afferent nucleus. Kn and kna index the nucleus name in nuclei and the position of this nucleus in the brain arrays. If kruse(kn)=0, we set it to 1 to show that this nucleus is active. If lw=3, we set kruse(kn) to 3 to identify it as a sensory nucleus. If lw=4, we set kruse(kn) to 4 to identify it as a motor nucleus.

If lx=2, we increment krnet(kn) and check it against knucwd for overflow. We place the values of kna, ly, lw, and lz in nucnet(kn,krnet(kn),__) where __ is 1, 2, 3, or 4.

If lx=3, we increment krgro(kn) and check it as we did for lx=2, and place the values of kna, ly, lw, and lz in nucgro.

The routine returns control to COLD.

```
      subroutine net
      include 'brain'
c
```

```
c this routine reads in the parameters of the neural net.
c
c if lx=1 this is a reservation only (a skeleton).
c      =2 synaptic instructions
c      =3 growth instructions
c
c if lx.eq.1 then
c        ly is the square root of the neurons to be reserved.
c        for 2 or 3, ly is the number of efferent synapses.
c if lx.eq.2 or lx.eq.3 then
c        ly is the number of efferent synapses with straight
c            (randomized possibly) topological connection.
c if ly.gt.10 then
c        the first digit codes the type of synaptic
c        distribution as in Hubel.
c
c if lw=3 this is a sensory nucleus
c      =4 this is a motor nucleus
c
c    lz=1 synapses are excitatory
c      =2 synapses are inhibitory
c      =3 contralateral excitatory
c      =4 contralateral inhibitory
c
   10 continue
      read(2,15,end=300) lx,ly,lw,lz,a,b
   15 format(4i5,2a30)
c
c here we get the number of the efferent nucleus
c
      call ident(a,jn,kn)
      if((jn.eq.0).or.(kn.eq.0)) then
      write(13,25) a
   25 format(1h ,'in net, ',a30,' not found in names')
      end if
      if(kruse(kn).eq 0) kruse(kn)=1
c
c the reservation of the square of ly is a skeleton and is
c negated here.
c
      if(lx.eq.1) go to 10
      if(lw.eq.3) kruse(kn)=3
      if(lw.eq.4) kruse(kn)=4
c
c here we get the number of the afferent nucleus
c
      call ident(b,jna,kna)
      if((jna.eq.0).or.(kna.eq.0))then
      write(13,25) b
      write(*,*) 'in net failed to find',b
      end if
      if(lx.eq.2) then
```

```
      krnet(kn)=krnet(kn)+1
      if(krnet(kn).gt.knucwd) krnet(kn)=knucwd
      nucnet(kn,krnet(kn),1)=kna
      nucnet(kn,krnet(kn),2)=ly
      nucnet(kn,krnet(kn),3)=lw
      nucnet(kn,krnet(kn),4)=lz
      go to 10
      end if
      if(lx.eq.3) then
      krgro(kn)=krgro(kn)+1
      if(krgro(kn).gt.knucwd) krgro(kn)=knucwd
      nucgro(kn,krgro(kn),1)=kna
      nucgro(kn,krgro(kn),2)=ly
      nucgro(kn,krgro(kn),3)=lw
      nucgro(kn,krgro(kn),4)=lz
      go to 10
      end if
      go to 10
  300 continue
      return
      end
```

CONNEX

We evaluate ly for class of connection and make a call either to TOPO for topological efferents or to HUBEL for filtered efferents. Following these calls, we call SYNNRM for each neuron to normalize the afferent synapses if it should be necessary.

The routine returns control to COLD.

Connections in the Cortex

The function of the cortex is to accept signal energy, primarily from the thalamus, and provide multiple paths to the caudate nucleus. We may think of these multiple paths as providing filtration, if we like. The signal energy turns on neurons in the cortex according to patterns, patterns that are not intuitive, not what we would expect. The incoming visual energy has already arranged itself in the retina in center-on and center-off patterns. These patterns consist first of a field, a region in visual space in which light will have an effect on the output neuron. The field has a center and a surround. The center excites and the surround inhibits.

On entering the koniocortex these center-surround signals arrange themselves in line segments. The next arrangement involves motion perpendicular to a line segment. The first represents a pattern in space; the second, a pattern in time of the spatial pattern.

The more complex patterns can only be simple combinations in space and time of the patterns in a previous column or area.

Since little may be said about these patterns beyond the very first ones, we will pretend that they are repetitions of the primary patterns. It is our most fundamental belief that they must be simple. We intend to add routines that will discriminate moving lines and then lines moving from or to the center. These will provide the neural analog of "things moving toward" and "things moving away."

```
      subroutine connex
      include 'brain'
c
c the kind of connection is looked at. it is either a filter or
c topographic at present.
c
      do 70 i=1,nrnucs
      if(kruse(i).eq.0) go to 70
      if(krnet(i).eq.0) go to 70
      do 60 j=1,krnet(i)
      if(nucnet(i,j,2).ge.10) then
      call hubel(i,j)
      else
      call topo(i,j)
      end if
   60 continue
   70 continue
c
c the afferent synapses are normalized.
c
      do 80 l=1,nrnucs
      do 80 k=1,kside
      do 80 j=1,kside
   80 call synnrm(j,k,l)
      return
      end
```

TOPO(lf,lnr)

This routine connects each neuron in the lfth nucleus to the equivalent neuron in the lnrth nucleus in its table of genetic connection. If there is to be more than one synapse per neuron, there is provision for random selection of neighboring neurons. A call to INSERT establishes the synapse. The routine then returns control to CONNEX.

We feel that this is a basic pattern of connection but there is always and accompanying filtration. The preservation of topology is a strong parameter in the genetic organization of a brain.

```
      subroutine topo(lf,lnr)
      include 'brain'
c
c lf is nucleus from, lnr is index of nucleus to
c
      apot=stinc
      ltime=1
      lt=nucnet(lf,lnr,1)
      ly=nucnet(lf,lnr,2)
      lw=nucnet(lf,lnr,3)
      lz=nucnet(lf,lnr,4)
c
c lw identifies the target nucleus as possibly sensory
c    or motor.
c ly is the number of synapses from to going.
```

```
c lz indicates type of synapse and whether ipsilateral
c    or contralateral.
c
      delta=1.0
      sigma=delta/2.36
      krdm=ly*delta**2
      if(krdm.lt.1) krdm=1
      do 40 kf=1,kside
      akf=kf
      do 30 jf=1,kside
      ajf=jf
      do 24 kqr=1,krdm
   27 continue
      kt=gauss(sigma,akf)
      jt=gauss(sigma,ajf)
      if((kt.lt.1).or.(kt.gt.kside)) go to 27
      if((jt.lt.1).or.(jt.gt.kside)) go to 27
      if(kqr.eq.1) then
      jt=jf
      kt=kf
      end if
      call insert(jf,kf,lf,jt,kt,lt,apot,ltime,lz)
      if(jt*kt*lt.eq.0) go to 27
   24 continue
   30 continue
   40 continue
      return
      end
```

HUBEL(1f,1nr)

This, like TOPO, is a genetic connection routine. Here we attempt a more significant distribution of efferent synapses. We use Hubel's directed line segments as our choice to add some realism to our intercortical connections. A call to INSERT establishes the synapse.

This routine effects a complex mapping from the efferent nucleus to the cortex. Each cortical neuron responds to a line segment. The segment itself has an inhibiting zone on both sides. We do not see this inhibiting effect occurring here, but in the previous stage. As we pass through the cortex we come to ever more complex cells, but the complexity is not in the immediate cell but in the previous cells. The inhibiting borders are a natural outcome of what came before: the center on, surround off cell.

The routine returns control to CONNEX.

```
      subroutine hubel(lf,lnr)
      include 'brain'
c
c lf is nucleus from, lnr is index of nucleus to
c
      dimension ktrial(16,16,16,2)
      apot=stinc
      !time=1
      lt=nucnet(lf,lnr,1)
```

```
      ly=nucnet(lf,lnr,2)
      lw=nucnet(lf,lnr,3)
c      nucnet(lf,lnr,4)=mod(nucnet(lf,lnr,4),10)   why this here?
      lz=nucnet(lf,lnr,4)
      do 150 jta=1,kside
      do 140 kta=1,kside
      do 130 ii=1,kside
      jt=jta
      kt=kta
      lt=nucnet(lf,lnr,1)
      jf=ktrial(ii,kt,jt,1)
      kf=ktrial(ii,kt,jt,2)
      call insert(jf,kf,lf,jt,kt,lt,apot,ltime,lz)
      if(jt*kt*lt.eq.0) go to 130
  130 continue
  140 continue
  150 continue
      return
      entry hublin
      aone=0.1989124
      atwo=0.414214
      athree=0.668179
      do 10 j=1,kside
      ktrial(j,1,1,1)=1
      ktrial(j,1,2,1)=(kside-j)*aone+1.5
      ktrial(j,1,3,1)=(kside-j)*atwo+1.5
      ktrial(j,1,4,1)=(kside-j)*athree+1.5
      ktrial(j,1,5,1)=kside-j+1
  10  continue
      do 20 l=1,5
      do 20 k=2,kside
      do 20 j=1,kside
      w=ktrial(j,k-1,l,1)+1
      if(w.gt.kside) w=w-kside
      ktrial(j,k,l,1)=w
  20  continue
      do 30 l=1,5
      do 30 k=1,kside
      do 30 j=1,kside
      ktrial(j,k,l,2)=j
  30  continue
      do 40 l=6,9
      la=10-l
      do 40 k=1,kside
      do 40 j=1,kside
      ktrial(j,k,l,1)=ktrial(j,k,1,2)
      ktrial(j,k,l,2)=kside-ktrial(kside-j+1,k,la,1)+1
  40  continue
      do 50 l=10,13
      la=18-l
      do 50 k=1,kside
      do 50 j=1,kside
```

```
      ktrial(j,k,l,1)=ktrial(j,k,1,2)
      ktrial(j,k,l,2)=kside-ktrial(j,k,la,2)+1
      if(l.eq.13) then
      w=ktrial(j,k,l,1)+k-1
      if(w.gt.kside) w=w-16
      ktrial(j,k,l,2)=w
      end if
   50 continue
      do 60 l=14,16
      la=kside-l+2
      do 60 k=1,kside
      do 60 j=1,kside
      ktrial(j,k,l,2)=ktrial(j,k,1,2)
      ktrial(j,k,l,1)=ktrial(kside-j+1,k,la,1)
   60 continue
c
c checkout coding follows
c
c     do 100 k=1,16
c     write(53,73)
c     write(53,73) k
c  73 format('set',i5)
c     write(53,73)
c     do 100 i=1,16
c     write(53,77) ((ktrial(i,j,k,l),j=1,16),l=1,2)
c  77 format(16i3,5x,16i3)
c 100 continue
c     close(53)
c     call exit
      return
      end
```

INSERT(jc,kc,lc,jd,kd,ld,apot,ltime,lz)

INSERT searches the list of afferent synapses on the desired neuron looking for a synapse with specific properties. If we find a match, we increment its potentiation by the value of apot; otherwise we establish a new synapse. The routine returns control to the caller.

```
      subroutine insert(jc,kc,lc,jd,kd,ld,apot,ltime,lz)
      include 'brain'
      logical g,h
c
c jc     row of efferent neuron
c kc     column of efferent neuron
c lc     efferent nucleus
c jd     row of afferent neuron
c kd     column of afferent neuron
c ld     afferent nucleus
c apot   potential of synapse to be established
c ltime  delay time of synapse
c lz=1   excitatory ipsilateral
c   =2   inhibitory ipsilateral
```

```
c   =3    excitatory contralateral
c   =4    inhibitory contralateral
c
      la=ksynct(jd,kd,ld)
c
c connection of up to 30 neurons allowed.
c multiple synapses are aggregated.
c
      if(la.le.0) go to 15
      do 10 kq=1,la
      g=ksynps(1,kq,jd,kd,ld).eq.jc
      g=g.and.(ksynps(2,kq,jd,kd,ld).eq.kc)
      g=g.and.(ksynps(3,kq,jd,kd,ld).eq.lc)
      g=g.and.(ksynps(4,kq,jd,kd,ld).eq.lz)
      g=(g.and.(ktime(kq,jd,kd,ld).eq.ltime))
      q=potent(kq,jd,kd,ld)
      if((lz.eq.1).or.(lz.eq.3)) then
      iop=1
      else
      iop=2
      end if
      if(lz.gt.2) apot=apot/4.0
      h=((iop.eq.1.and.q.ge.0.0).or.(iop.eq.2.and.q.le.0.0))
      if(g.and.h) then
      potent(kq,jd,kd,ld)=q+apot
      return
      end if
   10 continue
   15 continue
      if(la.ge.30) then
      jd=0
      kd=0
      ld=0
      return
      end if
      la=la+1
      ksynct(jd,kd,ld)=la
      ksynps(1,la,jd,kd,ld)=jc
      ksynps(2,la,jd,kd,ld)=kc
      ksynps(3,la,jd,kd,ld)=lc
      ksynps(4,la,jd,kd,ld)=lz
      potent(la,jd,kd,ld)=apot
      ktime(la,jd,kd,ld)=ltime
      synred(la,jd,kd,ld)=1.0
      return
      end
```

VISIIN

The next call by SETUP is to VISIIN an entry in VISUAL. VISIIN describes the division of visual space. The space extends 360 degrees horizontally, and from -20 to +50 vertically. We divide it into 256 regions for each eye.

```
      subroutine visual
      save
      include 'brain'
      dimension base(3,3),horiz(2,16),vert(2,16),ho(17),ve(17),
     1    wall(3,4),aa(3),bb(3),c(3),d(3),e(3),f(3),g(3)
      logical lck

      a='l lateral geniculate            '
      b='r lateral geniculate            '
      call ident(a,jx,kl)
      call ident(b,jx,kr)
      call space(base)
      high=0.5
      do 5 j=1,kside
      do 5 k=1,kside
      histry(1,j,k,kl)=0.0
      histry(1,j,k,kr)=0.0
    5 continue

      do 200 i=1,nrwall

      do 10 k=1,3
      wall(k,1)=awalls(k,1,i)
      wall(k,2)=awalls(k,1,i)
      wall(k,3)=awalls(k,2,i)
      wall(k,4)=awalls(k,2,i)
   10 continue
      do 20 k=2,3
      wall(2,k)=wall(2,k)+high
   20 continue
      do 25 l=1,3
      f(l)=0.5*(body(1,l,knrrad-1)+body(2,l,knrrad-1))
      aa(l)=wall(l,2)-wall(l,1)
      bb(l)=wall(l,4)-wall(l,1)
      d(l)=wall(l,1)
   25 continue
      call cross(aa,bb,c)
      do 50 j=1,kside
      do 50 k=1,kside
      do 50 m=1,2
      g(1)=horiz(2,k)*vert(2,16-j)
      g(2)=vert(1,16-j)
      g(3)=horiz(1,k)*vert(2,16-j)
c
c Whatever the meaning of left and right
c this will reverse the action.
c
      if(m.eq.2) g(3)=-g(3)
      do 30 ja=1,3
      e(ja)=0.0
      do 30 ka=1,3
      e(ja)=e(ja)+base(ka,ja)*g(ka)
```

```
30  continue
    call veccos(c,e,wans)
    if(abs(wans).lt.0.01745) then
    lck=.false.
    else
    call punctr(c,d,e,f,g)
    do 35 kw=1,3
    zq=(g(kw)-f(kw))*e(kw)
    if(zq.lt.-0.000001) then
    lck=.false.
    go to 37
    end if
35  continue
    call inout(g,4,wall.lck)
    end if
37  continue
    if(lck) then
    call dstnce(f,g,ans)
    if(ans.gt.0.01) then
    dans=1.0/ans
    else
    dans=100.0
    end if
    if(m.eq.1) then
    if(histry(1,j,k,kl).lt.dans) histry(1,j,k,kl)=dans
    else
    if(histry(1,j,k,kr).lt.dans) histry(1,j,k,kr)=dans
    end if
    end if
50  continue

200 continue

    return

    entry visiin
    ho(1)=0
    ho(2)=1
    ho(3)=2
    ho(4)=3
    ho(5)=4
    ho(6)=5
    ho(7)=7
    ho(8)=9
    ho(9)=11
    ho(10)=15
    ho(11)=25
    ho(12)=40
    ho(13)=68
    ho(14)=96
    ho(15)=124
    ho(16)=152
```

```
      ho(17)=180

      ve(1)=-20
      ve(2)=-10
      ve(3)=-5
      ve(4)=-4
      ve(5)=-3
      ve(6)=-2
      ve(7)=-1
      ve(8)=0
      ve(9)=1
      ve(10)=2
      ve(11)=3
      ve(12)=4
      ve(13)=5
      ve(14)=7
      ve(15)=10
      ve(16)=20
      ve(17)=50

      rads=6.28319/360.0

      do 530 k=1,kside
      hoz=rads*(ho(k+1)+ho(k))/2.0
      vez=rads*(ve(k+1)+ve(k))/2.0
      horiz(1,k)=sin(hoz)
      horiz(2,k)=cos(hoz)
      vert(1,k)=sin(vez)
      vert(2,k)=cos(vez)
  530 continue
      return
      end
```

MOTRIN

The last call by setup is to MOTRIN an entry in MOTR that sets parameters about the response of Pacrat to frustration.

We have now completed all the action under SETUP and return control to MAIN. MAIN calls LISTER.

```
      subroutine motr
      save
      include 'brain'
      ivade=ivade-1
      if(ivade.lt.0) ivade=0
      icont=icont-1
      if(icont.lt.0) icont=0
   10 if(ivade.gt.0) then
      call newdir(kdirec)
      go to 215
      end if
```

```
c
      a='l motor turnout          '
      call ident(a,j,k)
      bigl=0.0
      do 200 ka=1,kside
      do 200 kb=1,kside
      if(histry(1,ka,kb,k).gt.bigl) bigl=histry(1,ka,kb,k)
  200 continue
      a='r motor turnout          '
      call ident(a,j,k)
      bigr=0.0
      do 210 ka=1,kside
      do 210 kb=1,kside
      if(histry(1,ka,kb,k).gt.bigr) bigr=histry(1,ka,kb,k)
  210 continue
      if((bigl.ge.atriga).and.(bigr.ge.atriga))then
      ivade=70
      if(rand(x).gt.0.5) then
      kdirec=3
      else
      kdirec=4
      end if
      go to 10
      end if
      if((bigl.ge.atrig).or.(bigr.ge.atrig))then
      icont=5+2*rand(x)
      if(bigl.ge.bigr) then
      call newdir(3)
      write(38,*) icnt,'tactil went left'
      go to 215
      else
      call newdir(4)
      write(38,*) icnt,'tactil went right'
      go to 215
      end if
      else
      if(ivade.gt.0) go to 213
      wa=videol(x)
      wb=videor(x)
      if((wa+wb).gt.0.0) then
      if(icont.gt.0) go to 213
      if(wa.ge.wb) then
      call newdir(4)
      write(38,*) icnt,'vision went right'
      go to 215
      else
      call newdir(3)
      write(38,*) icnt,'vision went left'
      go to 215
      end if
      end if
      end if
```

```
 213 continue
     dxm=dxm*1.05
     if(dxm.gt.0.05) dxm=0.05
     go to 220
 215 if(dxm.gt.0.05) then
     dxm=dxm*0.9
     else
     dxm=dxm*0.95
     end if
     if(dxm.lt.0.02) dxm=0.02
c
 220 continue
     kckm=0
 222 continue
     avectr(1)=dxm
     phase=phase+5.0*dxm
     if(phase.gt.6.28319) phase=phase-6.28319
     call movpac
     if((dxm.lt.0.0).and.(kckm.eq.0))then
     kckm=1
     go to 222
     end if
c
     if(mod(icnt,10).eq.0) then
     write(4,240) icnt,knrrad
     240 format(2i10)
     do 250 k=1,knrrad
     do 250 i=1,2
     write(4,242) (body(i,j,k),j=1,3)
 242 format(3f10.5)
 250 continue
     end if
     return
     entry motrin
     ivade=0
     atrig=0.001
     atriga=atrig*20.0
     return
     end
```

LISTER

LISTER is a last step in preparing the brain to run. It writes out the names of the centers and of the nuclei. Then it makes a complete analysis of the net and also writes it to the file. It tabulates the efferent and afferent synapses of each nucleus. Following this recapitulation, we return control to MAIN for the last time before execution proper.

```
     subroutine lister
     include 'brain'
     dimension ktemp(1000,3,2)
     write(30,*) 'centers'
     write(30,*)
```

```
      do 5 k=1,nrctrs
 5    write(30,*) k,' ',centrs(k)
      write(30,*)
      write(30,*) 'nuclei'
      write(30,*)
      do 10 k=1,nrnucs
10    write(30,*) k,' ',nuclei(k)
      do 20 l=1,nrnucs
      if(kruse(l).eq.0) go to 20
      write(18,15) nuclei(l),
     1     alertd(l),runavg(l)
15    format(1h ,a30,5x,3f10.5)
20    continue
      do 100 mx=1,nrctrs
      write(18,35) centrs(mx)
35    format(1h0,' *** *** ',a30)
      do 90 l=kcnucs(mx,1),kcnucs(mx,2)
      if(kruse(l).eq.0) go to 90
      write(18,55) nuclei(l)
55    format(1h0,a30)
      write(18,57)
57    format(1h ,5x,'genesis')
      if(krnet(l).eq.0) then
      write(18,59)
59    format(1h ,16x,'nowhere')
      else
      do 70 kr=1,krnet(l)
      write(18,63) nucnet(l,kr,2),nuclei(nucnet(l,kr,1))
63    format(1h ,10x,i5,x,2a30)
70    continue
      end if
      write(18,65)
65    format(1h ,5x,'epigenesis')
      if(krgro(l).eq.0) then
      write(18,59)
      else
      do 80 kg=1,krgro(l)
      write(18,63) nucgro(l,kg,2),nuclei(nucgro(l,kg,1))
80    continue
      end if
      write(18,82)
82    format(1h ,5x,'genetic afferents from')
      kvinc=0
      do 85 la=1,nrnucs
      if(krnet(la).eq.0) go to 85
      do 84 lax=1,krnet(la)
      if(nucnet(la,lax,1).ne.l) go to 84
      kvinc=1
      write(18,63) nucnet(la,lax,2),nuclei(la)
84    continue
85    continue
      write(18,86)
```

```
 86 format(1h ,5x,'epigeneticafferents from')
    do 88 la=1,nrnucs
    if(krgro(la).eq.0) go to 88
    do 87 lax=1,krgro(la)
    if(nucgro(la,lax,1).ne.l) go to 87
    kvinc=1
    write(18,63) nucgro(la,lax,2),nuclei(la)
 87 continue
 88 continue
    if(kvinc.eq.0) write(18,59)
 90 continue
100 continue
    do 110 l=1,nrnucs
    do 110 ka=1,3
    do 110 la=1,2
110 ktemp(ja,ka,la)=0
    do 150 l=1,nrnucs
    if(kruse(l).eq.0) go to 150
    do 125 k=1,kside
    do 125 j=1,kside
    if(ksynct(j,k,l).eq.0) go to 125
    do 120 i=1,ksynct(j,k,l)
    ja=ksynps(1,i,j,k,l)
    ka=ksynps(2,i,j,k,l)
    la=ksynps(3,i,j,k,l)
    if(potent(i,j,k,l).ge.0.0) then
    ktemp(l,1,1)=ktemp(l,1,1)+1
    ktemp(la,2,1)=ktemp(la,2,1)+1
    if(l.eq.la) then
    ktemp(l,3,1)=ktemp(l,3,1)+1
    end if
    else
    ktemp(l,1,2)=ktemp(l,1,2)+1
    ktemp(la,2,2)=ktemp(la,2,2)+1
    if(l.eq.la) then
    ktemp(l,3,2)=ktemp(l,3,2)+1
    end if
    end if
120 continue
125 continue
150 continue
    do 190 l=1,nrnucs
    if(kruse(l).eq.0) go to 190
    write(18,185) nuclei(l),(ktemp(l,jn,1),jn=1,3)
185 format(1h ,a30,' afferents ',i5,' efferents ',
    1      i5,' self ',i5)
    write(18,187) (ktemp(l,jn,2),jn=1,3)
187 format(1h ,5x,'inhibiting',26x,i5,11x,i5,6x,i5)
190 continue
    return
    end
```

The Main Loop

We have built the brain, the organism, and the universe. The life of Pacrat begins.

REFERENCES

Evarts, E. V., Closing Remarks, in: *Functions of the Basal Ganglia* (D. Evered & M. O'Connor, Eds.), Pitman, London, 1984, pp. 269-272.

Hubel, D. H., *Eye, Brain, and Vision*, Freeman and Co., New York, 1988.

Jones, E. G., *The Thalamus*, Plenum Press, New York, 1985.

Kissin, B., *Conscious and Unconscious Programs in the Brain*, Plenum Press, New York, 1986.

Schneider, J. S., Opening Remarks, in: *Basal Ganglia and Behavior: Sensory Aspects of Motor Functioning*, (J. Schneider & T. Lidsky, Eds.), Hans Huber Publishers, Bern, 1985, pp. 1-8.

APPENDIX

```
 character*30 a,b,centrs,nuclei,nuclst
 logical chkout,hotstr,flags

 common/names/ nrctrs,nrnucs,centrs(100),nuclei(1000),
1   kcnucs(100,2),kruse(1000),krnet(1000),krgro(1000),
2   nucnet(1000,20,4),nucgro(1000,20,4),
3   kcensz,knucsz,knucwd,stinc,knucs,kside

 common/abrain/
1   ksynps(4,30,16,16,300),potent(30,16,16,300),
2   synred(30,16,16,300),ksynct(16,16,300),
3   histry(10,16,16,300),ktime(30,16,16,300)

 common/dynam/ alertd(1000), runavg(1000), thresh(16,16,300)

 common/plastk/ recptv(16,16,300),grfctr(16,16,300),
1   synare(2,2,16,16,300),grinc

 common/updatp/ reduce,frustr,frlvla,frlvlb,
1   rewavg,recovr,thrup,thrdwn,whz

 common/run/ chkout,hotstr,ksnap,flags(100),nuclst(100),
1   kbndls(3),lstnr(10),icnt,kbgend,kltend

 common/pacrt/ knrrad,radii(2,3,100),body(2,3,100),nrwall,
1   awalls(3,2,100)

 common/movng/ avectr(3),amoton(3,3),bmoton(3,3,100),
1   amovr(3,100),phase,radi(2,3,100)

 common/transf/ snpass(16,16,100),ampass(16,16,100),dist,
1   disr(2),dxm

 dimension alatgl(16,16),alatgr(16,16),venpl(16,16),
1   venpr(16,16),area4l(16,16),area4r(16,16)
 equivalence (alatgl,snpass),(alatgr(1),snpass(1,1,2)),
1          (venpl(1),snpass(1,1,3)),
2          (venpr(1),snpass(1,1,4)),
3          (area4l,ampass(1,1,3)),
4          (area4r(1),ampass(1,1,4))
```

a

a character variable available for general use.

alertd(1000) dynam
level of general activation of a nucleus.

amoton(3,3) movng
a movement matrix.

amovr(3,100) movng
a movement matrix.

ampass(16,16,100) transf
used to pass motor information from Pacrat to the universe.

avectr(3) movng
displacement of the center of Pacrat's nose.

awalls(3,2,100) pacrt
specifications of the walls.

b
a character variable available for general use.

bmoton(3,3,100) movng
a movement matrix.

body(2,3,100) pacrt
actual position of Pacrat's body.

centrs(100) names
names of the brain centers.

chkout run
flag that determines whether intermediate listings are desired.

dist(2) transf
temporary variables used in tactil for distance check.

dist transf
minimum distance from Pacrat to the closest wall.

dxm transf
the basic increment of movement.

flags(100) run
the flags that determine the combination of

intermediate printouts.

frlvla updatp
the upper frustration level check.

frlvlb updatp
the lower frustration level check.

frustr updatp
the frustration level (an emotion).

grfctr(16,16,300) plastk
the pressure on an axon to grow.

grinc plastk
the potentiation of a new synapse.

histry(10,16,16,300)abrain
activation history of a neuron.

hotstr run
flag for a hot start.

icnt run
counter for time steps.

kbgend run
the outer loop in the main program.

kbndls(3) run
three parameters for saving histories
(equivalent to do loop.)

kcensz names
upper limit on number of centers.

kcnucs(100,2) names
first and last nucleus in each center.

kltend run
inner loop in main program.

knrrad pacrt

number of body segments.

knucs names

total number of nuclei.

knucsz names

limit on number of nuclei.

knucwd names

upper limit on number of nuclei this nucleus may be efferent upon - or grow to.

krgro(1000) names

the actual number of nuclei this nucleus may grow to.

krnet(1000) names

the actual number of nuclei this nucleus is genetically efferent upon.

kruse(1000) names

type of nucleus (sensor, motor, other).

kside names

the number of neurons on one side of a nucleus (they are square.)

ksnap run

interval for snapshots (mod(icnt,ksnap))

ksynct(16,16,300) abrain

count of synapses on this neuron.

ksynps(4,30,16,16,3)abrain

description of afferent neuron: 1 row, 2 column, 3 nucleus, 4 type.

ktime(30,16,16,300)abrain

axonal delay to synapse.

lstnr(10) run

number of nuclei to be listed in this

printout.

nrctrs names
the number of centers.

nrnucs names
the number of nuclei.

nrwall pacrt
the number of walls.

nucgro(1000,20,4) names
growth: 1 nucleus to, 2 type of
distribution, 3 type of neuron, 4 type of synapse.

nuclei(1000) names
names of nuclei.

nuclst(100) run
names of nuclei to be listed by CHEKPR.

nucnet(1000,20,4) names
genetic: 1 nucleus to, 2 type of
distribution, 3 type of neuron, 4 type of synapse.

phase movng
the phase of the sinusoidal movement of
Pacrat.

potent(30,16,16,300)abrain
potentiation of a synapse.

radi(2,3,100) movng
working position of Pacrat.

radii(2,3,100) pacrt
absolute position of Pacrat.

recovr updatp
increment of dynamic synaptic condition.

recptv(16,16,300) plastk
receptivity of a neuron to the formation of
synapses.

reduce updatp
decrement of dynamic synaptic condition.

rewavg updatp
used in coding for inhibitory growth.

runavg(1000) dynam
running average of nuclear activity.

snpass(16,16,100) transf
used to pass sensory information from the universe to Pacrat.

stinc names
original potentiation of a genetic synapse.

synare(2,2,16,16,30)plastk
synaptic area: from this nucleus and from others.

synred(30,16,16,300)abrain
readiness of this synapse.

thrdwn updatp
hypopolarization increment.

thresh(16,16,300) dynam
dynamic threshold of this neuron.

thrup updatp
hyperpolarization increment.

whz updatp
used in hillok and in normal.

TECHNICAL REPORT INTERNAL DISTRIBUTION LIST

	NO. OF COPIES
CHIEF, DEVELOPMENT ENGINEERING DIVISION	
ATTN: SMCAR-CCB-DA	1
-DC	1
-DI	1
-DR	1
-DS (SYSTEMS)	1
CHIEF, ENGINEERING SUPPORT DIVISION	
ATTN: SMCAR-CCB-S	1
-SD	1
-SE	1
CHIEF, RESEARCH DIVISION	
ATTN: SMCAR-CCB-R	2
-RA	1
-RE	1
-RM	1
-RP	1
-RT	1
TECHNICAL LIBRARY	5
ATTN: SMCAR-CCB-TL	
TECHNICAL PUBLICATIONS & EDITING SECTION	3
ATTN: SMCAR-CCB-TL	
OPERATIONS DIRECTORATE	1
ATTN: SMCWV-ODP-P	
DIRECTOR, PROCUREMENT DIRECTORATE	1
ATTN: SMCWV-PP	
DIRECTOR, PRODUCT ASSURANCE DIRECTORATE	1
ATTN: SMCWV-QA	

NOTE: PLEASE NOTIFY DIRECTOR, BENET LABORATORIES, ATTN: SMCAR-CCB-TL, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST

	NO. OF COPIES
ASST SEC OF THE ARMY RESEARCH AND DEVELOPMENT ATTN: DEPT FOR SCI AND TECH THE PENTAGON WASHINGTON, D.C. 20310-0103	1
ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN: DTIC-FDAC CAMERON STATION ALEXANDRIA, VA 22304-6145	12
COMMANDER US ARMY ARDEC ATTN: SMCAR-AEE	1
SMCAR-AES, BLDG. 321	1
SMCAR-AET-O, BLDG. 351N	1
SMCAR-CC	1
SMCAR-CCP-A	1
SMCAR-FSA	1
SMCAR-FSM-E	1
SMCAR-FSS-D, BLDG. 94	1
SMCAR-IMI-I (STINFO) BLDG. 59 PICATINNY ARSENAL, NJ 07806-5000	2
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-DD-T, BLDG. 305 ABERDEEN PROVING GROUND, MD 21005-5066	1
DIRECTOR US ARMY MATERIEL SYSTEMS ANALYSIS ACTV ATTN: AMXSY-MP ABERDEEN PROVING GROUND, MD 21005-5071	1
COMMANDER HQ, AMCCOM ATTN: AMSMC-IMP-L ROCK ISLAND, IL 61299-6000	1
COMMANDER ROCK ISLAND ARSENAL ATTN: SMCRI-ENM ROCK ISLAND, IL 61299-5000	1
DIRECTOR US ARMY INDUSTRIAL BASE ENGR ACTV ATTN: AMXIB-P ROCK ISLAND, IL 61299-7260	1
COMMANDER US ARMY TANK-AUTMV R&D COMMAND ATTN: AMSTA-DDL (TECH LIB) WARREN, MI 48397-5000	1
COMMANDER US MILITARY ACADEMY ATTN: DEPARTMENT OF MECHANICS WEST POINT, NY 10996-1792	1
US ARMY MISSILE COMMAND REDSTONE SCIENTIFIC INFO CTR ATTN: DOCUMENTS SECT, BLDG. 4484 REDSTONE ARSENAL, AL 35898-5241	2
COMMANDER US ARMY FGN SCIENCE AND TECH CTR ATTN: DRXST-SD 220 7TH STREET, N.E. CHARLOTTESVILLE, VA 22901	1
COMMANDER US ARMY LABCOM MATERIALS TECHNOLOGY LAB ATTN: SLCMT-IML (TECH LIB) WATERTOWN, MA 02172-0001	2

<u>NOTE:</u> PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST (CONT'D)

	NO. OF COPIES
COMMANDER US ARMY LABCOM, ISA ATTN: SLCIS-IM-TL 2800 POWDER MILL ROAD ADELPHI, MD 20783-1145	1
COMMANDER US ARMY RESEARCH OFFICE ATTN: CHIEF, IPO P.O. BOX 12211 RESEARCH TRIANGLE PARK, NC 27709-2211	1
DIRECTOR US NAVAL RESEARCH LAB ATTN: MATERIALS SCI & TECH DIVISION CODE 26-27 (DOC LIB) WASHINGTON, D.C. 20375	1 1
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-IB-M (DR. BRUCE BURNS) ABERDEEN PROVING GROUND, MD 21005-5066	1

	NO. OF COPIES
COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MN EGLIN AFB, FL 32542-5434	1
COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MNF EGLIN AFB, FL 32542-5434	1
MIAC/CINDAS PURDUE UNIVERSITY 2595 YEAGER ROAD WEST LAFAYETTE, IN 47905	1

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET LABORATORIES, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.